Radial Basis Function of Neural Network in Performance Attribution

WONG HING-KWOK

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Statistics

June 2003

Abstract of thesis entitled:

Radial Basis Function of Neural Network in Performance Attribution

Submitted by Wong Hing Kwok

for the degree of Master of Philosophy in Statistics

at The Chinese University of Hong Kong in June 2003.

Abstract

It is well known that linear models cannot adequately represent the complex nature of financial data. Using traditional linear techniques in performance attribution can be very misleading as demonstrated in the recent findings of Chan and Genovese (2001). In this thesis, a class of nonlinear models, namely, radial basis function of neural networks, is proposed to evaluate the performance attribution problem of a financial portfolio. By constructing by a universe of stocks using specific factors, prediction of excess returns of the stocks from the constructed portfolio is studied and compared with those resulted from using multifactor linear models. It is found that the radial basis function approach outperforms the linear model approach and compares favorably with the additive model approach proposed in Chan and Genovese. Furthermore, due to the flexibility of the radial basis function approach and the neural network setting, further enhancements can be gained by tuning the network. The thesis concludes with a discussion and comparison of using different nonlinear and basis function approaches in the study of performance attribution.

摘要

線性模式在應用於複雜的財務資料上，是不足以完全解釋變數之間的關係。而近期 Chan 和 Genovese (2001) 發現傳統的線性模式在應用於績效歸屬上，會有誤導的產生，所以研究一種能正確估計績效歸屬的非線性模式是有必要的。這篇論文，將會評估神經網路中的輻射形基底函數於財務組合的績效歸屬中的應用成效。

首先在大量的股票中，利用輻射形基底函數的方法，把股票的超額報酬以特定的因子去估計出來，而根據這估計的超額報酬，建立一個股票組合。再以多因子線性模式的方法去，建立其他股票組合。之後比較各組合的回報。發現輻射形基底函數方法比線性模式方法好，也比 Chan 和 Genovese 所建議的累進模式方法毫不遜色。再者，由於輻射形基底函數方法的設定彈性很大，所以利用網路的轉變能提升輻射形基底函數方法在績效歸屬的表現。而論文的結尾會討論及比較不同的非線性模型及基本函數應用在績效歸屬的效能。

Acknowledgement

I would like to express to my sincere gratitude to my supervisor, Prof. CHAN Ngai- hang, for his helpful suggestions to this research program. It has been a great privilege to work with him. I have not only improved my ability in thinking, but also enhanced my research ability. I would also like to express my thanks to my fellow classmates and all the staff of the Statistics Department.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Performance attribution is an active subject in the financial literature. It attempts to explain the return of a portfolio at a given time. Therefore, performance attribution can be used to validate a manager's investment process. For such an attribution, security returns are usually assumed to be related to prespecified factors and sector factors.

Linear multifactor model is one of the common models used in performance attribution. It is used for a special class of relationship between measurable quantities that can be described by a generalization of a straight line in higher dimension. In performance attribution, linear multifactor model establishes a linear relationship between the excess returns of a portfolio or an index with a number of factors, such as market capitalization, book value, book-to-market ratio, and earning surprise. The model is of the form

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon, \tag{1.1}$$

where y is the excess returns (dependent variable), x_i is the explanatory factor i (independent variables), ϵ is the random error and β_i is the regression coefficient that represents the linear relationship between the excess returns and the explanatory factors i. Although linear regression provides a rich set of models for analysis, as financial data are after complicated, using simple linear method to explain the complicated relationship between excess returns and specific factors may not be appropriate. It results in low explanatory power. One way to resolve this problem is to entertain nonlinear relationships between returns and factors. Usually, a nonlinear model is of the form

$$y = f(x_1, x_2, \cdots, x_p) + \epsilon, \tag{1.2}$$

where $f(x_1, x_2, \cdots, x_p) =$ is an unknown nonlinear function that represents the nonlinear relationship between the excess returns and the explanatory factors. For example, if $f(x_1, x_2, \cdots, x_p) = \beta_0 + \beta_1 {x_1}^2 + \beta_2 {x_2}^2 + \cdots + \beta_p {x_p}^2$, then there is a quadratic relationship between returns $\{y\}$ and factors $\{x_i\}$. Nonlinear regression can improve the explanatory power and the accuracy for predictions. Nonlinear techniques in performance attribution also outperforms the linear techniques, as demonstrated in Chan and Genovese (2001).

Neural network is one of the nonlinear techniques proposed in Chan and Genovese (2001). The discipline of neural network is very broad. It involves sophisticated concepts and tools from mathematics, theoretical physics, to designs of VLSI chip so that fast computational systems can be constructed for a wide range of applications, see L.J.Landau and J.G.Taylor (1997). Since the emergence of

the simplest kind of neural network, neural network has been evolved into other generations such as multiplayer feedforward network (MFN) and recurrent network. Radial basis function (RBF) network is one of the latest generation of neural network that has been developed in recent years. It has gained increasing popularity due to its desirable properties in classification and functional approximation, accompanied by the fact that training of RBF is more rapid than many other neural network techniques. The RBF network is proving to be an useful tool in several areas: including robotics, biomedical engineering, and finance. For example, Hutchinson, Lo and Poggio (1994) demonstrated the usefulness of RBF network in the pricing of derivatives.

In this thesis, an RBF network is proposed to evaluate the performance attribution problem of a portfolio. Using specific factors from a fixed universe of stocks, predictions of excess returns of the stocks from the constructed portfolio is studied and comparisons with multifactor linear models are conducted. There are two major reasons for choosing RBF network as our study objective. First, it is adaptive and responsive to structural changes in the data-generating processes. Second, it is flexible enough to encompass a wide range of derivative securities and fundamental asset price dynamics, see Hutchinson, Lo and Poggio (1994).

This thesis is organized as follows. Chapter 2 describes the RBF network in detail. RBF models are defined and estimation methods are described. In Chapter 3, RBF networks are applied in performance attribution for a financial data set. A brief introduction of linear and additive approach is given in Chapter 4. Comparisons between RBF and other approaches are also given in this chapter.

Concluding remarks are given in Chapter 5.

Chapter 2

Radial Basis Function (RBF) of Neural Network

The emergence of neural networks is an attempt to analyze the functionality of the human brain. In the view of an engineer or a computer scientist, a neural network can be characterized most adequately as a 'computational model' with particular properties, such as the ability to learn, to adapt, and to generalize. The structure of a neural networks has been developed from the simplest form into more advanced types such as the multiplayer feedforward network (MFN) and the recurrent network. RBF network is a neural network that has been developed in recent years. It is widely used in different fields of research such as statistical science, engineering, computer science, and artificial intelligence, see Ando, Konish and Imoto(2002). In this chapter, an introduction to the neural network and the RBF network is given in Sections 2.1 and 2.2. Section 2.3 describes the detail of our model while Section 2.4 describes estimation methods.

2.1 Neural Network

A neural network is constructed by a number of highly interconnected processors (units, nodes, and neurons), which are the analogs of biological neural cells in a human brain. A number of weighted links are connected to the neurons allowing signals to pass through them. Each neuron typically receives input signals from a number of incoming connections. After receiving the input signals, the neuron produces a single outgoing signal which passes through the neuron's connection (corresponding to the biological axon of a neuron). The outgoing connection usually splits into a number of branches. Each branch of the single outgoing connection transmits the same signal to different places. For example, some of the branches end at the incoming connection of other neurons in the network, and some may end outside the network to generate controls or response patterns. These connecting rules are directly inspired by the structure of a human brain. Figure 2.1 shows a typical neural network with one single layer.

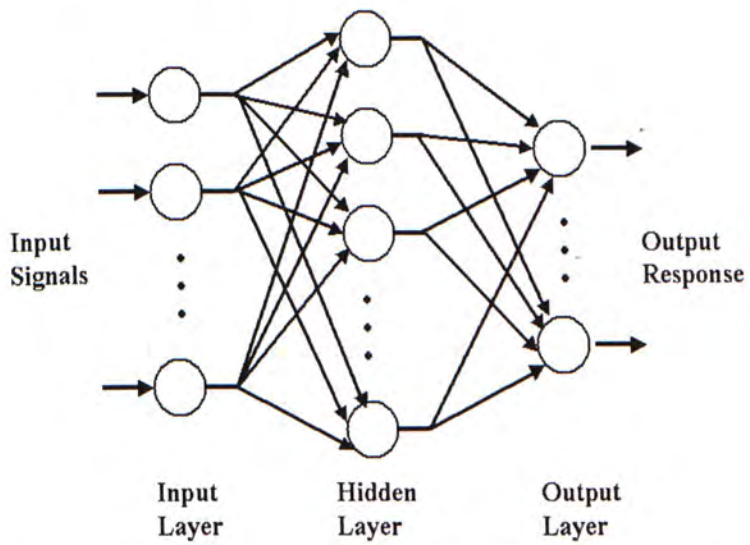

Figure 2.1: Typical Neural Network with One Single Layer

For the time being, the mathematical expression of the neuron is simply a mapping function. It maps a weighted combination of inputs to an output value. It usually chooses a 'sigmoidal' nonlinear function (a monotonic increasing function from zero to one) with the form

$$0 \leq f(a) \leq f(b) \leq 1, \quad \text{for} \quad a \leq b, \tag{2.1}$$

as the mapping function. The neural network can approximate a large class of nonlinear functions after combining such simple units with multiple interconnections. Figure 2.2 is an example of a neural network, called a feed forward neural network, with a single layer.

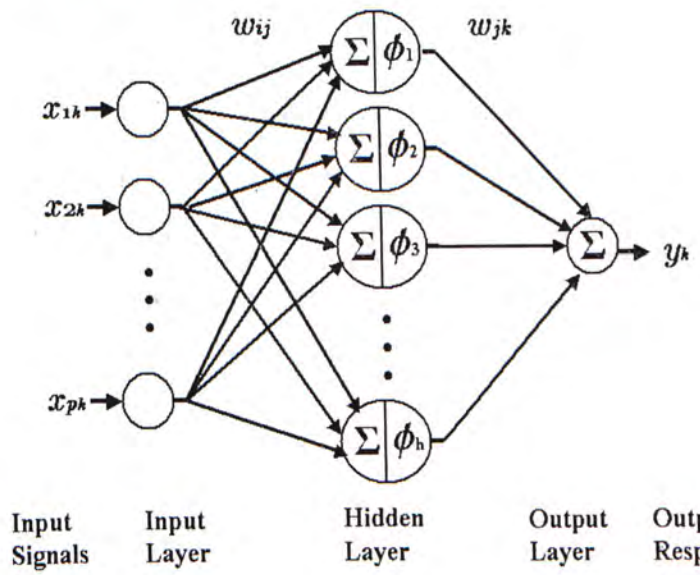

Figure 2.2: Feed Forward Neural Network with a Single Layer

It can be written as a nonlinear function in the form

$$y_k = \sum_{j=1}^{h} w_{jk} \phi_j \left(\sum_{i=1}^{p} w_{ij} x_{ik} \right), \tag{2.2}$$

where y_k is the k^{th} output value; x_{ik} is the i^{th} input value of the output value; w_{ij} (w_{jk}) is the interconnection weight from unit i (j) in the input (hidden) layer

to unit j (k) in the hidden (output) layer, and $\phi_j(\cdot)$ is the sigmoidal nonlinear function defined in equation (2.1) of the j^{th} hidden unit.

Many model can be consider as a neural network, for example, choosing the sigmoidal function $\phi_j(x) = x$ and adding a 'skip-layer', w_{0k} providing a direct links between the input and output layers. Then the function of equation (2.2) becomes a linear model with the form

$$y_k = w_{0k} + \sum_{i=1}^{p} w_{ik} x_{ik}, \tag{2.3}$$

2.2 Radial Basis Function (RBF) Network

An RBF network is a single hidden layer feedforward network. It is conceptually simple and it is capable of modeling any nonlinear function. In particular, an RBF network can be considered as a multiple nonlinear regression.

Suppose we have a data set $\{(\boldsymbol{x_k}, y_k);\ \text{for } k = 1, 2, \ldots, n\}$, where $\boldsymbol{x_k}$ is a p-dimensional vector of independent variables and y_k is a dependent variable. Then a nonlinear regression model is of the form

$$y_k = f(\boldsymbol{x_k}) + \epsilon_k, \tag{2.4}$$

where $f(\cdot)$ is an unknown nonlinear function and ϵ_k is the error term with mean zero ($\mathrm{E}(\epsilon_k|\boldsymbol{x_k}) = 0$). The unknown function $f(\cdot)$ is estimated by minimizing the objective function

$$H(\hat{f}) \equiv \sum_{k=1}^{n} \left(\|y_k - \hat{f}(\boldsymbol{x_k})\|^2 + \lambda \|P\hat{f}(\boldsymbol{x_k})\|^2 \right), \tag{2.5}$$

where λ is a constant, $\hat{f}(\cdot)$ is an approximation of $f(\cdot)$, $P\hat{f}(\boldsymbol{x_k}) = (\frac{\partial \hat{f}(\boldsymbol{x_k})}{x_1}, \frac{\partial \hat{f}(\boldsymbol{x_k})}{x_2}$ $,\cdots, \frac{\partial \hat{f}(\boldsymbol{x_k})}{x_n})'$ is the differential operator and $\|\cdot\|$ is a vector norm. A vector norm is a scalar that gives some measure of the magnitude of the elements of the vector and there are many different types of vector norm, for example, infinity norm, Frobenius-norm and Euclidean norm.

The objective function $H(\hat{f})$ is the sum of two parts. The first part measures the distance between the approximation $\hat{f}(\boldsymbol{x_k})$ and the observation y_k, the second term is a penalty function which will decrease when $\hat{f}(\cdot)$ is more smooth, and λ controls the tradeoff between fitting and smoothness.

Under certain conditions, see Poggio and Girosi (1990), the function that minimizes the objective function $H(\hat{f})$ is an RBF of the form

$$\hat{f}(\boldsymbol{x_k}) = \sum_{j=1}^{h} w_j \phi_j \left(\|\boldsymbol{x_k} - \boldsymbol{c_j}\|\right) + p(\boldsymbol{x_k}), \tag{2.6}$$

where $\boldsymbol{c_j}$ is a p-dimensional vector determining the center of the RBF for the j^{th} unit, w_j is the final layer weights connecting the j^{th} hidden node to the output node, $\phi_j(\cdot)$ is the nonlinear transfer function of the j^{th} hidden node, $p(\boldsymbol{x_k}) = \alpha_0 + \boldsymbol{\alpha_1}'(\boldsymbol{x_k}) + \boldsymbol{\alpha_2}'(x_{k1}^2, x_{k2}^2, \ldots, x_{kp}^2) + \cdots$ is a polynomial operator, and x_{ki} is the factor i of $\boldsymbol{x_k}$.

From equation (2.6), it can be seen that the hidden nodes provide a set of "function" $\phi_j(\cdot)$ that constitutes a "basis" for the network input. The output of the RBF network is a linear combination of the outputs from its hidden layer,as stipulated by the right hand side of (2.6). This structure demonstrates that an RBF network allows for a much simpler weight updating procedure and can be

easily described by a set of nonlinear equations. Figure 2.3 illustrates the physical connections of the RBF network.

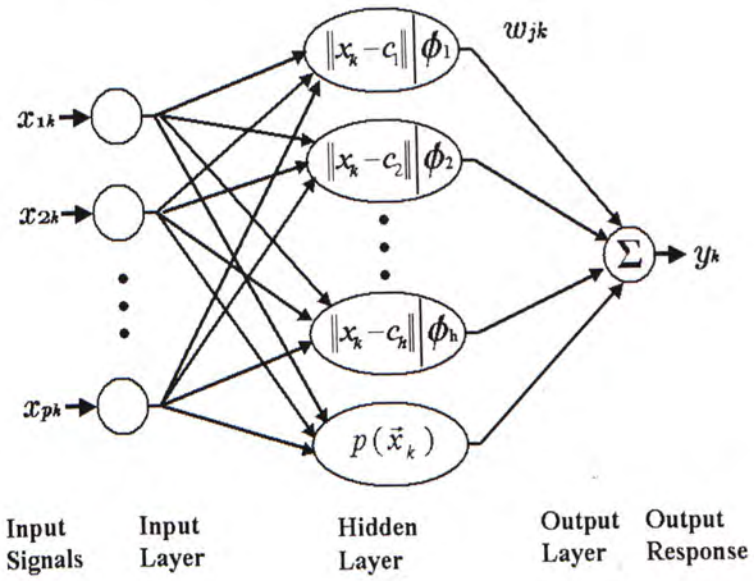

Figure 2.3: Radial Basis Function Network

2.3 Model Specification

To identify the model, one has to choose the number of hidden nodes h, the degree of the polynomial operator $p(\cdot)$, the vector norm $\|\cdot\|$, and the nonlinear transfer function of the hidden node $\phi_j(\cdot)$, which is usually selected from a class of basis functions as follows:

(a) Gaussian,

$$\phi(x, \sigma) = \exp(-x^2/\sigma^2), \tag{2.7}$$

(b) Multi-quadratic function,

$$\phi(x, \sigma) = (x^2 + \sigma^2)^{1/2}, \tag{2.8}$$

(c) Inverse multi-quadratic function,

$$\phi(x, \sigma) = (x^2 + \sigma^2)^{-1/2}, \tag{2.9}$$

(d) Thin plate spline,

$$\phi(x) = -x^2 \log(x), \tag{2.10}$$

where σ is a scaling parameter.

Choosing the number of hidden nodes h poses a difficult task, since the number of network parameters (for example the centers $\{\boldsymbol{c_j}\}$ and the connection weights $\{w_j\}$) are directly related to the number of hidden nodes h. Similar to multi-linear regressions, simple models (corresponding to a small h in our case) would have high approximation errors but low estimation errors, while complex models (corresponding to a large h in our case) would have low approximation errors but high estimation errors. There exists no simple answer to find the right trade-off between approximation and estimation errors. Niyogi and Girosi (1994) introduce an optimal choice of h that minimizes the generalization error for any fixed number of data n. The generalization error $E[(f_o - \hat{f}_h)^2]$, where f_o is the 'optimal' solution of the function or the true function and $\hat{f}_h$ is the approximation function with h hidden nodes, is used to measure the distance between the true function and the approximation function. They suggest $h = n^{1/3}$ as the number of hidden nodes, which is very similar to the optimal solution of h that minimizes the generalization error for any fixed number of data n.

For simplicity, we let all hidden nodes have the same transfer function but with different parameters ($\phi_j(x) = \phi(x, \sigma_j)$; for $j = 1, 2, \ldots, h$). We use a Gaussian

and a multi-quadratic function for hidden node $\phi(\cdot)$. The degree of the polynomial operator $p(\cdot)$ is chosen to be zero and one, and $\|\cdot\|$ is chosen as the Euclidean norm defined by $\|\boldsymbol{x_k}\| = (\sum_{i=1}^{n} {x_i}^2)^{1/2}$. The RBF model becomes

Model 1 When degree of $p(\cdot)$ is zero and a Gaussian transfer function is used,

$$y_k = \sum_{j=1}^{h} w_j \exp\left(-\|\boldsymbol{x_k} - \boldsymbol{c_j}\|^2/\sigma_j^2\right) + \alpha_0 + \epsilon_k. \tag{2.11}$$

Model 2 When degree of $p(\cdot)$ is zero and a multi-quadratic transfer function is used,

$$y_k = \sum_{j=1}^{h} w_j \left(\|\boldsymbol{x_k} - \boldsymbol{c_j}\|^2 + \sigma_j^2\right)^{1/2} + \alpha_0 + \epsilon_k. \tag{2.12}$$

Here α_0 is the coefficients of the polynomial $p(\cdot)$ a Gaussian transfer function is used.

Model 3 When degree of $p(\cdot)$ is one and ,

$$y_k = \sum_{j=1}^{h} w_j \exp\left(-\|\boldsymbol{x_k} - \boldsymbol{c_j}\|^2/\sigma_j^2\right) + \alpha_0 + \boldsymbol{\alpha_1}'(\boldsymbol{x_k}) + \epsilon_k. \tag{2.13}$$

Model 4 When degree of $p(\cdot)$ is one and a multi-quadratic transfer function is used,

$$y_k = \sum_{j=1}^{h} w_j \left(\|\boldsymbol{x_k} - \boldsymbol{c_j}\|^2 + \sigma_j^2\right)^{1/2} + \alpha_0 + \boldsymbol{\alpha_1}'(\boldsymbol{x_k}) + \epsilon_k, \tag{2.14}$$

Here $\boldsymbol{\alpha_1} = (\alpha_1, \alpha_2, \ldots, \alpha_p)'$ are the coefficients of the polynomial $p(\cdot)$.

2.4 Estimation

An important task of regression is the estimation of unknown parameters. For a given data set with n sets of p-dimensional inputs, $\{\boldsymbol{x_k}\}$, and outputs,

$\{y_k\}$, the total number of parameters to be estimated for models 1 and 2 is $(p+2)h+1$. It includes ph elements of the centers $\{\boldsymbol{c_j}\}$, h coefficients in $\{\sigma_j\}$, h parameters in weights $\{w_j\}$, and one polynomial coefficient α_0. The total number of parameters to be estimated for the models 3 and 4 is $(p+2)h+p+1$, which includes p additional polynomial coefficients in $\boldsymbol{\alpha_1}$.

A common way to estimate these RBF parameters is to apply nonlinear estimation methods such as the Levenberg-Marquardt method (see Marquardt (1963)). When the model is complicated, such as those used in high dimensional problems or non-convex optimization problems, local minima may occur. In such cases, stochastic optimization methods may be used, but they are usually very time-consuming. A full account of these methods can be found in Hutchinson (1993).

Alternatively, a different approach is described by Ando, Konish and Imoto (2002). It divides the estimation process into two stages. In the first stage, the centers $\{\boldsymbol{c_j}\}$ and the parameters $\{\sigma_j\}$ are determined by using the inputs $\{\boldsymbol{x_k}\}$ only. In the second stage, the weight parameters $\{w_j\}$ of the output layer and the coefficients of the polynomial α_0 and $\boldsymbol{\alpha_1}$ are estimated by regular estimation methods such as the least-squares or the maximum likelihood. One of the advantages is that the computational burden of this approach is much smaller than traditional nonlinear estimation methods. In this thesis, the estimation method is based on the Ando, Konish and Imoto (2002) approach but using a generalized least-squares (GLS) in the second stage. GLS is a widely used estimation method in regression. It is more general then the weighted least-squares (WLS) and the

ordinary least-squares (OLS). GLS is applicable when the error ϵ is correlated and heteroskedastic (nonconstant variance), while WLS is applicable only when the error ϵ is uncorrelated, and OLS is applicable when the error ϵ is both uncorrelated and homoscadestic (constant variance). In fact, OLS and WLS can be considered as special cases of GLS. GLS is more appropriate for this study because the information of the error is unknown in the study. It is more general to assume that the error is correlated and heteroskedastic (nonconstant variance). Even the error is not correlated and heteroskedastic, GLS is still applicable. Discussions of GLS can be found in Weisberg (1985).

Specifically, in the first stage, a k-mean clustering algorithm is used to determine the centers $\{\boldsymbol{c_j}\}$ and the parameters $\{\sigma_j\}$. This algorithm partitions the input data set $\{\boldsymbol{x_k};\ \text{for } k = 1, 2, \ldots, n\}$ into h clusters $\{A_j;\ \text{for } j = 1, 2, \ldots, h\}$, each cluster corresponds to one hidden node. Each cluster contains approximately the same number of data points. The centers $\{\boldsymbol{c_j}\}$ and the parameters $\{\sigma_j\}$ are calculated by

$$\boldsymbol{c_j} = \frac{1}{n_j} \sum_{\boldsymbol{x_\alpha} \in A_j} \boldsymbol{x_\alpha}, \qquad {\sigma_j}^2 = \frac{1}{n_j} \sum_{\boldsymbol{x_\alpha} \in A_j} \|\boldsymbol{x_\alpha} - \boldsymbol{c_j}\|, \tag{2.15}$$

where n_j ($n_1 = \cdots = n_{h-1} = \lfloor n/h \rfloor$, $n_h = n - (h-1)\lfloor n/h \rfloor$) is the number of data that belongs to the j^{th} cluster A_j and $\|\cdot\|$ is the Euclidean norm.

In the second stage, GLS is used to estimate the weights $\{w_j\}$ of the output layer and the polynomial coefficients α_0 and $\boldsymbol{\alpha_1}$. Consider models 1 and 2. They can be expressed in a matrix form as

$$\boldsymbol{y} = \boldsymbol{\Phi}(\mathbf{X})\boldsymbol{w} + \boldsymbol{\epsilon}, \tag{2.16}$$

where $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)'$ is the vector of output data, $\boldsymbol{w} = (w_1, w_2, \ldots, w_h, \alpha_0)'$ is the vector that contains the wiegths $\{w_j\}$ and the polynomial coefficients α_0, $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)'$ is the vector of errors with mean zero, and variance covariance matrix $\sigma^2\boldsymbol{\Sigma}$. In other words, $E(\boldsymbol{y}) = \boldsymbol{\Phi}(\mathbf{X})\boldsymbol{w}$ and the variance covariance matrix of $\boldsymbol{y}$ is $\sigma^2\boldsymbol{\Sigma}$. Here $\boldsymbol{\Phi}(\mathbf{X})$ is the matrix of transformed inputs

$$\boldsymbol{\Phi}(\mathbf{X}) = \begin{bmatrix} \phi(\|\boldsymbol{x_1} - \boldsymbol{c_1}\|, \sigma_1) & \cdots & \phi(\|\boldsymbol{x_1} - \boldsymbol{c_h}\|, \sigma_h) & 1 \\ \phi(\|\boldsymbol{x_2} - \boldsymbol{c_1}\|, \sigma_1) & \cdots & \phi(\|\boldsymbol{x_2} - \boldsymbol{c_h}\|, \sigma_h) & 1 \\ \vdots & & \vdots & \vdots \\ \phi(\|\boldsymbol{x_n} - \boldsymbol{c_1}\|, \sigma_1) & \cdots & \phi(\|\boldsymbol{x_n} - \boldsymbol{c_h}\|, \sigma_h) & 1 \end{bmatrix},$$

where $\phi(x, \sigma_j)$ is the Gaussian or multi-quadratic function. By minimizing the Residual Sum of Squares (RRS):

$$\min_{\boldsymbol{w}} (\boldsymbol{y} - \boldsymbol{\Phi}(\mathbf{X})\boldsymbol{w})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}(\mathbf{X})\boldsymbol{w}), \tag{2.17}$$

the GLS estimator is given by

$$\hat{\boldsymbol{w}} = (\boldsymbol{\Phi}(\mathbf{X})'\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}(\mathbf{X}))^{-1}(\boldsymbol{\Phi}(\mathbf{X})'\boldsymbol{\Sigma}^{-1}\boldsymbol{y}). \tag{2.18}$$

For models 3 and 4, the only difference is that the matrix of transformed inputs $\boldsymbol{\Phi}(\mathbf{X})$ has the form,

$$\boldsymbol{\Phi}(\mathbf{X}) = \begin{bmatrix} \phi(\|\boldsymbol{x_1} - \boldsymbol{c_1}\|, \sigma_1) & \cdots & \phi(\|\boldsymbol{x_1} - \boldsymbol{c_h}\|, \sigma_h) & 1 & x_{11} & \cdots & x_{1p} \\ \phi(\|\boldsymbol{x_2} - \boldsymbol{c_1}\|, \sigma_1) & \cdots & \phi(\|\boldsymbol{x_2} - \boldsymbol{c_h}\|, \sigma_h) & 1 & x_{21} & \cdots & x_{2p} \\ \vdots & & \vdots & \vdots & \vdots & \vdots \\ \phi(\|\boldsymbol{x_n} - \boldsymbol{c_1}\|, \sigma_1) & \cdots & \phi(\|\boldsymbol{x_n} - \boldsymbol{c_h}\|, \sigma_h) & 1 & x_{n1} & \cdots & x_{np} \end{bmatrix},$$

where $\phi(x, \sigma_j)$ is the Gaussian or multi-quadratic function. And $\boldsymbol{w} = (w_1, w_2, \ldots, w_h, \alpha_0, \alpha_1, \ldots, \alpha_p)'$.

Moreover, equation (2.18) shows that if $\mathbf{\Sigma}$ is a diagonal matrix, then the GLS estimator will become a WLS estimator. If $\mathbf{\Sigma}$ is an identity matrix, then the GLS estimator will become an OLS estimator.

In next chapter, the estimation method will be used to fit the RBF model (models 1 to 4) with a financial data set and the RBF model in performance attribution will be evaluated.

Chapter 3

RBF in Performance Attribution

In this chapter, a financial data set is used to examine the merits of applying RBF in performance attribution. Our procedure of evaluating the RBF mimics the way of performance attribution proposed in Chan and Genovese (2001). Each month, the data set is used to fit the model and to predict the excess returns. Then, the predicted excess returns are used to rank the stocks and a portfolio is constructed based on these ranks. The portfolio is maintained over the study period, longing and shorting in cap-weighted proportion as needed at the recorded prices.

In this chapter, the background of the data set is described in Section 3.1. Sections 3.2 and 3.3 give the detail of portfolio construction and performance evaluation. The result is given in Section 3.5.

3.1 Background of Data Set

The data used in this study consists of roughly 1,000 large-cap US stocks (similar to consituents in the Frank Russell 1000 index) over a two-year period from November 1998 though October 2000. The data was recorded on a monthly basis about excess returns, prices, cap-weights, and several explanatory factors as follows:

Dependent Variable:

y_t - Excess returns (Market residual return): Return that are independent of the benchmark, that is,

$$y_t = r_t - \beta \tilde{r}_t \tag{3.1}$$

where r_t is a difference between asset return and riskless rate, β is the measure of an asset's risk in relation to the market (for example, the S&P500) or to an alternative benchmark or factors, $\tilde{r}_t$ is a difference the between benchmark return and riskless rate, and the benchmark is the performance of a predetermined set of securities and used for comparison purposes.

Independent Variables:

$x_{1,t}$ - Book-to-price: Ratios based on annual book values and current prices.

$x_{2,t}$ - Return-on-equity: Forward-looking estimate of return-on-equity. This is determined from dividing the net income for the past 12 months by common stockholder equity.

$x_{3,t}$ - Dispersions of Institutional Brokers Estimate System (I/B/E/S) consensus earning forecasts. A measure of future earnings uncertainty (inverted scale). I/B/E/S is a database of earnings expectations data obtained from more than 2,500 security analysis. The consensus earnings estimates for more than 3,000 companies are available with a separate subscription agreement. All of I/B/E/S historical estimate databases are fully adjusted for splits and other capitalization changes to allow users to run longer data series applications.

$x_{4,t}$ - Changes in forecast earnings (I/B/E/S consensus) over the last month.

$x_{5,t}$ - Revisions Up: The percentage of earnings estimates revised upward over last month.

$x_{6,t}$ - Revisions Down: The percentage of earnings estimates revised downward over last month.

$x_{7,t}$ - Earnings Surprise: A measure of the gap between the reported quarterly earnings and the last relevant I/B/E/S consensus earnings estimate.

$x_{8,t}$ - Measure of projected earnings yield: It is the ratio of forecast earnings to prices.

Cap-weights are used to weight the stocks as follows:

κ - Cap-weight: One of the weighting methods used in computing a market index. Cap-weight of stock k is defined as

$$\kappa_k = \frac{(p_k \times s_k)}{\sum_{\alpha \in U} p_\alpha \times s_\alpha}, \tag{3.2}$$

where p_k is the prices of stock k in the index, s_k is the respective number of outstanding shares of stock k and U contains stocks of the wholes universe.

Before conducting this study, some data have been processed. The excess returns have been preprocessed to remove sector means prior to analysis. When data are missing due to merging or dropping out of business, they are removed from the study.

3.2 Portfolio Construction

At the beginning of each month t in the study period, explanatory factors $\{x_{ki,t-1}\}$ (independent variables) and cap-weight $\{\kappa_{k,t-1}\}$ (weighting) at the end of the month $t-1$ and the excess returns $\{y_{k,t}\}$ (dependent variable) at the end of month t are used to fit the RBF network models (model 1 to 4). The fitted model is then used with the factors $\{x_{ki,t}\}$ at the end of month t to predict the excess returns $\{\hat{y}_{k,t+1}\}$ for month $t+1$. Use model 1 be an example, for month t, the factors $\{x_{ki,t-1}\}$ at the end of the month $t-1$ are first separated into h clusters $\{A_j;\ \text{for } j = 1, 2, \ldots, h\}$ each cluster corresponds one hidden node and then put into formula (2.15), such that,

$$\begin{aligned} \boldsymbol{c_{j,t-1}} &= \frac{1}{n_j} \sum_{\boldsymbol{x_{\alpha,t-1}} \in A_j} \boldsymbol{x_{\alpha,t-1}}, \\ {\sigma_{j,t-1}}^2 &= \frac{1}{n_j} \sum_{\boldsymbol{x_{\alpha,t-1}} \in A_j} \|\boldsymbol{x_{\alpha,t-1}} - \boldsymbol{c_{j,t-1}}\|, \end{aligned}$$

where $\boldsymbol{x_{k,t-1}} = (x_{k1,t-1}, x_{k2,t-1}, \ldots, x_{k8,t-1})$ is the vector containing the factors at the end of the month $t-1$ of stock k, n_j is the number of data that belongs to

the j^{th} cluster A_j and $\|\cdot\|$ is the Euclidean norm. Then by using formula (2.18)

$$\hat{\boldsymbol{w}}_t = (\boldsymbol{\Phi}(\mathbf{X_{t-1}})'\boldsymbol{\Sigma_{t-1}}^{-1}\boldsymbol{\Phi}(\mathbf{X_{t-1}}))^{-1}(\boldsymbol{\Phi}(\mathbf{X_{t-1}})'\boldsymbol{\Sigma_{t-1}}^{-1}\boldsymbol{y_t}),$$

where $\hat{\boldsymbol{w}}_t = (\hat{w_{1,t}}, \hat{w_{2,t}}, \ldots, \hat{w_{h,t}}, \hat{\alpha}_{0,t})'$ is the vector that contain the estimated wiegths $\{w_j\}$ and the estimated polynomial coefficients α_0, $\boldsymbol{\Sigma_{t-1}}^{-1} = \text{diag}(\kappa_{1,t-1}, \kappa_{2,t-1}, \ldots, \kappa_{n,t-1})$, $\boldsymbol{y_t} = (y_{1,t}, y_{2,t}, \ldots, y_{n,t})'$ is the vector of excess returns over month t and $\boldsymbol{\Phi}(\mathbf{X_{t-1}})$ is the matrix of transformed factors $\{x_{ki,t-1}\}$

$$\boldsymbol{\Phi}(\mathbf{X_{t-1}}) = \begin{bmatrix} \phi(\|\boldsymbol{x_{1,t-1}} - \boldsymbol{c_{1,t-1}}\|, \sigma_{1,t-1}) & \cdots & \phi(\|\boldsymbol{x_{1,t-1}} - \boldsymbol{c_{h,t-1}}\|, \sigma_{h,t-1}) & 1 \\ \phi(\|\boldsymbol{x_{2,t-1}} - \boldsymbol{c_{1,t-1}}\|, \sigma_{1,t-1}) & \cdots & \phi(\|\boldsymbol{x_{2,t-1}} - \boldsymbol{c_{h,t-1}}\|, \sigma_{h,t-1}) & 1 \\ \vdots & & \vdots & \vdots \\ \phi(\|\boldsymbol{x_{n,t-1}} - \boldsymbol{c_{1,t-1}}\|, \sigma_{1,t-1}) & \cdots & \phi(\|\boldsymbol{x_{n,t-1}} - \boldsymbol{c_{h,t-1}}\|, \sigma_{h,t-1}) & 1 \end{bmatrix},$$

with $\phi(x, \sigma_j)$ being the Gaussian function defined in (2.7). By substituting the factors $\{x_{ki,t}\}$ at the end of month t into the fitted model, the vector of predicted excess returns for month $t+1$ is given by

$$\hat{\boldsymbol{y}}_{t+1} = \boldsymbol{\Phi}(\mathbf{X_t})\hat{\boldsymbol{w}}_t,$$

where $\boldsymbol{\Phi}(\mathbf{X_t})$ is the matrix of transformed factors $\{x_{ki,t}\}$.

After predicting the excess returns, a set of stocks is selected from the universe for buying or shorting according to their predicted excess returns. One approach for selecting stocks for entry into the portfolio is taking long positions on the top decile of the predicted excess returns. Another approach is taking long positions on the top decile and short positions on the bottom decile of the predicted excess returns. Both of these approaches are used and the portfolio is rebalanced. The process is then repeated in the next month until the end of the study period.

3.3 Portfolio Rebalance

For the long only approach, we first sell all the stocks $\{s_{k,t-1}\}$ held at the beginning of month t and use the proceeds ($c_t = \sum_k s_{k,t-1} \times p_{k,t}$, where $p_{k,t}$ is the prices of stock k at the beginning of month t) to buy the stocks $\{s_{k,t}\}$ in the top decile in cap-weighted proportion at prices $\{p_{k,t}\}$ at the beginning of month t, that is,

$$s_{k,t} = (\frac{c_t}{p_{k,t}})(\frac{\kappa_{k,t}}{\sum_{\gamma \in D_t} \kappa_{\gamma,t}}), \tag{3.3}$$

where D_t is the set containing the stocks in the top decile and $\kappa_{k,t}$ is the cap weight of the stock k at the end of month t.

For the long-short approach, we first repay the loan ($=\sum_k ss_{k,t-1} \times p_{k,t}$, where $ss_{k,t-t}$ is the stocks that are shorted in month $t-1$) with stocks certificates that are shorted in month $t-1$ and sell all the stocks $\{s_{k,t-1}\}$ held at the prices $\{p_{k,t}\}$ in month t. The amount becomes

$$c_t = \sum_k s_{k,t-1} \times p_{k,t} - \sum_k ss_{k,t-1} \times p_{k,t} + c_{t-1}, \tag{3.4}$$

where c_{t-1} is the remaining balance at the end of month $t-1$. Then, half of this amount, $c_t/2$, is used to buy the selected stocks in cap-weighted proportion at the prices of month t, that is,

$$s_{k,t} = (\frac{c_t}{2p_{k,t}})(\frac{\kappa_{k,t}}{\sum_{\gamma \in D_t} \kappa_{\gamma,t}}), \tag{3.5}$$

where D_t is the set containing stocks in the top decile and $\kappa_{k,t}$ is the cap weight of the stock k at the end of month t. At the same time, we short the selected

stocks in cap-weighted with total value equaling $c_t/2$. That is,

$$ss_{k,t} = (\frac{c_t}{2p_{k,t}})(\frac{\kappa_{k,t}}{\sum_{\gamma \in B_t} \kappa_{\gamma,t}}), \tag{3.6}$$

where B_t is a set that contain the stocks in the bottom decile and $\kappa_{k,t}$ is the cap weight of the stock k at the end of month t. The remaining half of cash $c_t/2$ is reserved in the portfolio corresponding to the cost of the shorted stocks $ss_{k,t}$. Then, total remaining amount is still c_t. The process is then repeated in the next month until the end of the study period.

In the case that a particular ticker is not available in next month, we set the purchase price equaling to the selling price of that ticker; yielding no gain or loss for moth t.

3.4 Result

Results are given as average monthly returns in percentage. The average monthly returns is defined as $100(r^{1/24} - 1)$ where r is the total return over the study period. The total return is the ratio of the final value of the portfolio to the initial value of the portfolio. Figure 3.1 shows the average monthly returns of the RBF network models (models 1 to 4) for the long only approach against the number of hidden nodes h, ranging from 1 to 50.

Form Figure 3.1, when the number of hidden nodes h is small (< 10), models 1 and 2 have much lower average monthly returns than models 3 and 4. When the number of hidden nodes h increases (> 10), the average monthly returns of models 1 and 2 increase to the levels of models 3 and 4. On the average, models

3 and 4 (where $p(\cdot)$ has first degree) perform better than models 1 and 2 (where $p(\cdot)$ has degree zero). The highest average returns of models 1 to 4 are 6.00%, 6.01%, 5.72% and 5.88% when h equals to 30, 34, 43, and 39 respectively.

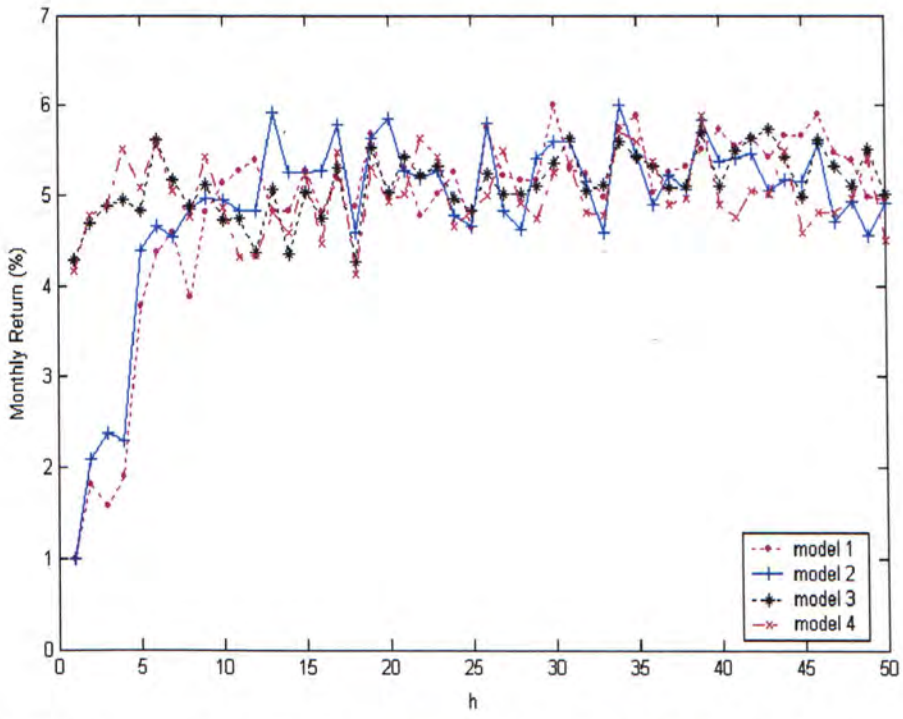

Figure 3.1: The average monthly returns of the models 1 to 4 for the long only approach against the number of hidden nodes h ranging from 1 to 50.

Figure 3.2 shows the average monthly returns of the RBF network models (models 1 to 4) for the long-short approach against the number of hidden nodes h ranging from 1 to 50. The result is similar to the long only approach. The highest average returns of models 1 to 4 are 6.28%, 6.33%, 5.96% and 6.19% when h equals to 30, 29, 46, and 30 respectively.

From Figure 3.1 and Figure 3.2, observe that the average monthly returns one affected by the number of hidden nodes h. Choosing the number of hidden nodes h is very important for the performance of RBF network models. As described in

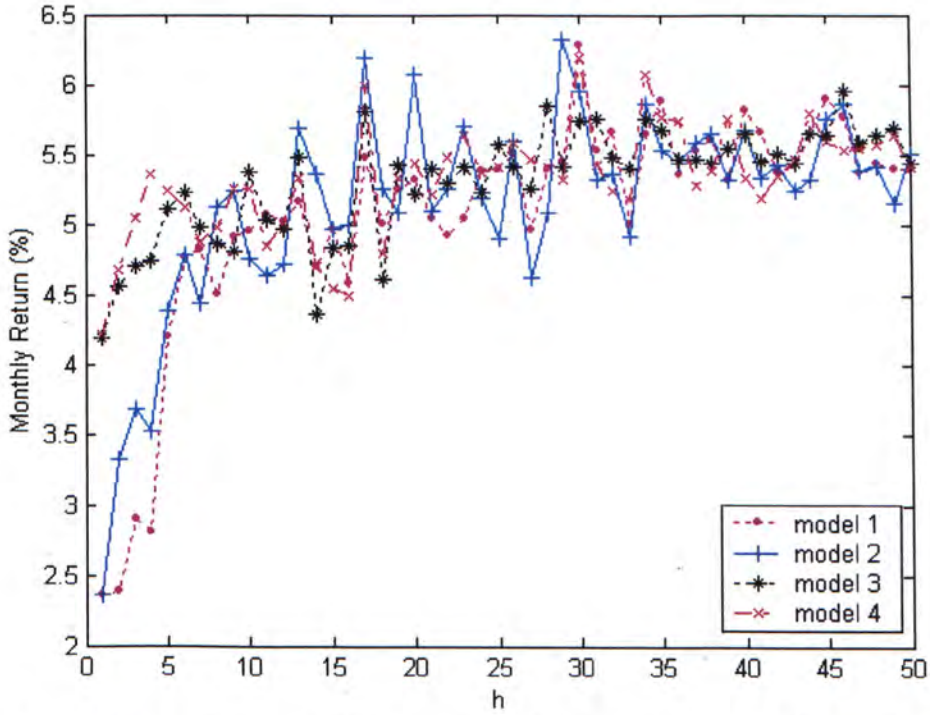

Figure 3.2: The average monthly returns of the model 1 to 4 for the long-short approach against the number of hidden nodes h ranging from 1 to 50.

Section 2.3, Niyogi and Girosi (1994) suggest $h = n^{1/3}$ as the number of hidden nodes. In our case, n lies between 898 and 1070, so the suggested h should be 9 or 10. However, the average monthly returns of four models with h equal 9 or 10 is low. It seems that the value of h suggested by Niyogi and Girosi (1994) does not work in this study. Result in this study suggests that h should be around 30 to 35 which is about $n^{1/2}$, which is a little bigger than the value of h suggested by Niyogi and Girosi (1994).

Chapter 4

Comparison

In the previous chapter, the RBF network is applied in performance attribution to a real data set. The preceding result shows a positive value of average returns. This means that a gain is achieved when the RBF network is applied. It will be difficult, however, to conclude that RBF is a superior technique in performance attribution as it has not been compared with other methods. To evaluate the performance of RBF in performance attribution, a comparison of the RBF with other models using linear or nonlinear techniques is conducted in this chapter. The compared models are standard linear model, fixed additive model, and the refined additive model proposed in Chan and Genovese (2001).

Standard linear model is chosen for comparison because it is one of the most commonly used models in performance attribution. Fixed additive models and refined additive model are chosen because they perform well according to Chan and Genovese (2001). Sections 4.1 to 4.3 give a brief introduction for these models. The result of the comparison is given in Section 4.4.

4.1 Standard Linear Model

The standard linear model is one of the most commonly used models in regression. It is widely used in performance attribution. A standard linear model assumes that excess returns (dependent variable) are a linear combination of explanatory factors (independent variables). The model is of the matrix form

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{4.1}$$

where $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)'$ is the vector of excess returns, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_8)'$ is the vector that contains the regression coefficient, $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)'$ is the error vector with mean zero, and variance $\boldsymbol{\Sigma}$, and $\mathbf{X}$ is the matrix contains the explanatory factors such that

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{18} \\ 1 & x_{21} & x_{22} & \cdots & x_{28} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{n8} \end{bmatrix},$$

where x_{ki} is the factor i of stock k. GLS estimation is used to fit the model. As discussed in section 2.4, the GLS estimator of the unknown parameters $\boldsymbol{\beta}$ is given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}(\mathbf{X}'\boldsymbol{\Sigma}^{-1}\boldsymbol{y}). \tag{4.2}$$

It minimizes the residual sum of squares,

$$\min_{\boldsymbol{\beta}}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}), \tag{4.3}$$

where $\mathbf{\Sigma^{-1}} = \text{diag}(\kappa_1, \kappa_2, \ldots, \kappa_n)$ and κ_k is the cap weight of stock k. As $\mathbf{\Sigma^{-1}}$ is a diagonal matrix, the GLS estimator $\hat{\boldsymbol{\beta}}$ in fact is a WLS estimator.

4.2 Fixed Additive Model

A fixed additive model extends the notion of a linear model by allowing some or all linear functions of the independent variables to be replaced by arbitrary smooth functions of the independent variables. An additive model is of the form

$$y_k = \mu + \sum_{i=1}^{8} f_i(x_{ki}) + \epsilon_k, \tag{4.4}$$

where μ is a constant, $f_i(\cdot)$ are unknown functions, and ϵ_k is the idiosyncratic error term for stock k.

The Gauss-Seidel method backfitting algorithm, which smoothes partial residuals iteratively is used to fit an additive model, see T.J.Hastie and R.J.Tibshirani (1990) for further details. To start the algorithm, initial values and functions are chosen with $\mu = \text{average}(\boldsymbol{y})$, and $f_i = \hat{f}_i^{\,0}$, for $i = 1, 2, \ldots, 8$ where $\hat{f}_i^{\,0}$ is an arbitrary smoothing spline. Then,

$$\hat{f}_i^{\,t+1} = S_i(\boldsymbol{y} - \mu - \sum_{l \neq i}(\hat{f}_l^{\,t})|\boldsymbol{x}_l), \tag{4.5}$$

where $S_i(\cdot)$ is a smoothing spline, $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)'$ is the vector of excess returns, and $\boldsymbol{x}_l = (x_{1l}, x_{2l}, \ldots, x_{nl})'$ is the vector of factors $l$of stocks 1 to n. The process keeps cycling (for $i = 1, 2, \ldots, p$) until the functions, f_i, $i = 1, 2, \ldots, 8$ converge. The algorithm separates the parametric part from the nonparametric part of the fit, and fits the parametric part using weighted linear least squares

within the backfitting algorithm.

4.3 Refined Additive Model

A refined additive model is defined as applying a model selection technique on a fixed additive model. The stepwise selection method is used a in refined additive model.

To understand the process, a fixed additive model (equation (4.5)) is fitted and is used as the initial model in the stepwise search. At each iteration in the stepwise selection, the Bayesian information criterion (BIC) of the current model is calculated, as well as those for all reduced and augmented models, then adding or dropping the term (linear or nonlinear) that reduces BIC the most as needed. The BIC is of the form

$$\text{BIC(model)} = -2\log(L(\text{model})) + p * \log(n), \tag{4.6}$$

where $L(\cdot)$ is the maximized likelihood function, p is a constant represents the number of parameters (including the intercept) and n the number of observations in the model.

The process repeats until no more reduction of the BIC is attained. Note that multifactor interactions are not included in the model. Finally, the model of the last iterative is the refined additive model and is fitted by the backfitting algorithm.

BIC has been widely used for model identification in time series and linear regression. It provides a method for determining the optimal trade-off between

model complexity and the model's ability to accurately represent the data. It has been shown that when $n \to \infty$, the model selected by BIC method converges to the correct model, see Schwarz G. (1978).

4.4 Result

Table 4.1 shows average monthly returns of the models in Sections 4.1 - 4.3 as well as model 1 to 4 for both the long only and the long-short re-balancing strategies. It also contains the corresponding average monthly return that is obtained by selling all the stocks held and then buying the entire universe of stocks in cap-weighted proportions each month.

Table 4.1: Summary of Average Monthly Returns of Each Models

Models	Average Monthly Returns 11/1998 though 10/2000	
	Long Top Decile	Long Top/ Short Botttom Decile
Standard Linear Model	4.33%	4.24%
Fixed Additive Model	5.10%	6.12%
Refined Additive Model	5.08%	6.12%
Model 1	6.00%($h = 30$)	6.28%($h = 30$)
Model 2	6.01%($h = 34$)	6.33%($h = 29$)
Model 3	5.72%($h = 43$)	5.96%($h = 46$)
Model 4	5.88%($h = 39$)	6.19%($h = 30$)
Whole Universe		-.89%

Observe that the RBF model shows a significant improvement in average

monthly returns over the linear model in both long only and long-short strategies. Comparing to the fixed additive model and the refined additive model, the RBF performs much better for the long only strategy. The RBF (excluding model 3) performs slightly better than other models for the long-short strategy. Note that, the average monthly returns of models 1 to 4 shown in Table 4.1 are the best cases among all h ranging from 1 to 50. In fact, for some of the non-optimal value of h models 1 to 4 perform much worse than other methods. As there seems to be no general method for choosing the number of hidden nodes h, it will be difficult to argue that RBF is better than additive models in general. A more prudent approach is to compare all the cases (h ranges from 1 to 50) of RBF models. In Figure 3.1 (long only strategy), it is seen that the average monthly returns are higher than that of the fixed and refined additive models where h is large (> 10). For the long-short strategy (Figure 3.3), only a few cases of h have average monthly returns bigger than that of the fixed model and the refined additive model. In other words, RBF models perform better in the long only strategy for most of the large h (> 10), but perform worse than the additive models in the long-short strategy for most h.

Chapter 5

Conclusion

In this thesis, an RBF network is proposed to evaluate the performance attribution problem of a portfolio. The finding in this thesis suggests that RBF network model can be a valuable approach in performance attribution.

When compare with the standard linear model, the RBF network approach performs significantly better. Although the RBF model is more complicated and requires more demanding computationally, the striking improvement shown in Figure 3.1 shows that the extra efforts used in fitting the RBF are well spent.

On the other hand, the RBF approach compares favorably with the additive model approach. The RBF model outperforms the additive model for some particular values of h, but performs worse than additive models in general. Since there is no general method of choosing the optimal h and since good softwares are available for fitting additive models, it is suggested that using additive models approach for performance attribution may be more appropriate.

Due to the flexibility of the RBF network approach in particular and the neural

network approach in general, further enhancements may be achieved by tuning the network. For example, increasing the number of hidden nodes h and the degree of $p(\cdot)$ may improve the performance of RBF network approach. This observation is revealed in the results shown in Chapter 3. However, increasing both h and the degree of $p(\cdot)$ is equivalent to increasing the number of parameters to be estimated. This will increase estimation errors and computational time. Also, the transfer function of the hidden nodes constitutes an important part of the RBF models; the structure of the RBF network depends on the transfer function. In this study, we use a fixed transfer function for all nodes h. For example, either a Gaussian or a multi-quadratic function has been used. The reason of choosing these two functions is because they are regarded as one of the most convenient basis functions in RBF models. In fact, instead of using a fixed transfer function for all nodes, we can let the transfer function of some hidden nodes to be different ($\phi_i(x) \neq \phi_j(x)$; for some $i \neq j$). In such cases, improvements maight be attained. Moreover, other estimation methods such as the Levenberg-Marquardt method (see Marquardt (1963)), may also lead to further improvements.

Bibliography

T. Ando, S. Konishi and S. Imoto, "Nonlinear regression modeling via regularized radial basis function networks," *Research Memorandum #845*, The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106-8569, Japan, July 2002.

N. H. Chan and C. R. Genovese, "A comparison of linear and nonlinear statistical techniques in performance attribution," *IEEE Transactions on Neural Networks*, vol. 12, pp. 922-928, July 2001.

T. J. Hastie and R. J. Tibshirani, *Generalized Additive Models*, Chapman and Hall, London, 1990.

J. M. Hutchinson, "A radial basis function approach to financial time series analysis," *Technical Report #1457*, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1993.

J. M. Hutchinson, A. W. Lo and T. Poggio, "A nonparametric approach to pricing and hedging derivative securities via learning networks," *The Journal of Finance*, vol. 49, pp. 851-889, July 1994.

L. J. Landau and J. G. Taylor, *Concepts for neural networks: a survey*, vol. , pp. , 1997.

D. W. Marquardt *J. Soc. Ind. Appl. Math.*, pp.431-441 , 1963.

P. Niyogi and F. Girosi, "On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions," *Artificial Intelligence memo #1467*, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, February 1994.

T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, pp. 1481-1497, September 1990.

G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics*, vol. 6, pp. 461-464, 1978.

S. Weisberg, *Applied Linear Regression*, Second Edition, Wiley, New York, 1985.